REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
7/20/99

3. REPORT TYPE AND DATES COVERED
Final, 4/20/95-4/19/99

4. TITLE AND SUBTITLE
"Extending Database Integration Technology"

5. FUNDING NUMBERS
DAAH04-95-1-0169

6. AUTHOR(S)
Peter Buneman

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Pennsylvania
Department of Computer & Information Science
200 S 33rd Street
Philadelphia, PA 19104-6389

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
US Army Research Office
PO Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
ARO 33159.6-MA

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this reportare those of the author adn should not be construed as an official US Army policy or decision.

12a. DISTRIBUTION AVAILABILITY STATEMENT
approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Formal approaches to the semantics of databases and database languages can have immediate and practical consequences in extending database integration technologies to include a vastly greater range of data sources and data structures. We consider three broad areas --- collection types, schema transformation, and partial information --- that are central to obtaining interoperability of heterogeneous data sources. In each of these areas we have developed working prototypes that have been put to practical use. This proposal describes work on collection types and schema transformation, and outlines a plan for the development of both principles and implementation of practical languages and tools that will extend database integration technology well beyond its current confines to cope with legacy systems, structured files, data-intensive applications, and other non-standard data sources.

14. SUBJECT TERMS

20001120 131

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Final Report on ARO Contract DAAH 04-95-1-0169

"Extending Database Integration Technology"

P. Buneman, S, Davidson and V. Tannen

The contract has partially supported the principal investigators and several PhD students.

Schema integration and transformation The need to transform data between heterogeneous databases arises from a number of critical tasks in data management. These problems are further complicated by schema evolution in the underlying databases, and by the presence of non-standard database constraints.

Davidson and Kosky describe a declarative language, WOL, for specifying such transformations, and an implementation, Morphase, based on this language. WOL is designed to allow transformations between the complex data structures which arise in object-oriented databases, as well as complex relational databases, and to allow for reasoning about the interactions between database transformations and constraints [21].

Kosky, Davidson and Buneman [1] discuss database transformations arising in many different settings including database integration, evolution of database systems, and implementing user views and data-entry tools. They also consider the problem of insuring the correctness of database transformations. In particular, we demonstrate that the usefulness of correctness conditions such as information preservation are hindered by the interactions of transformations and database constraints, and the limited expressive power of established database constraint languages.

Semantics of collection types Relying on previous work [3, 2] with R. Subrahmanyam, and S. Naqvi (Bellcore) Buneman and Tannen have identified primitives based on instances of structural recursion on collections. Category theory served us to understand the central role played by a particular instance: monad primitives. Together with L. Wong, we were able to propose and exploit a partial foundation to programming with collections in query languages [9, 4]. Buneman, Libkin, Suciu, Tannen and Wong continued the study of the use of collection comprehensions in database programming languages. The syntax of comprehensions is very close to the syntax of a number of practical database query languages and is, they believe, a better starting point than first-order logic for the development of database languages[9, 8].

In collaboration with the Penn bioinformatics group this has in turn led to a system for information integration, Kleisli, that was specialized to molecular biology data sources, with significant practical impact [11, 5]. Davidson, Hara, and Popa have further extended the query system Kleisli to provide an interface to the Shore object-oriented database system [10].

While collection restructuring (eg. the nested relational algebra) was nicely explained by the framework in [9, 4] aggregate operations on collections, collection constructors, and conversions between different kinds of collections were not. The monoid comprehension calculus of Fegaras and Maier provided such an approach. Together with K. Lellahi of University of Paris 13, Tannen was able to propose a more general approach, based on monad algebras and on a new robust notion of "enrichment" for monads [12]. Using this foundation, Tannen has designed the

core of our second-generation information integration system, K2, currently developed in our Penn Center for Bioinformatics.

Focusing on another collection type, Libkin, Machlin and Wong have developed an array query language and optimization techniques [13].

Semi-structured data A new kind of data model has recently emerged in which the database is not constrained by a conventional schema. Systems like ACeDB, which has become very popular with biologists, and the recent Tsimmis proposal for data integration organize data in tree-like structures whose components can be used equally well to represent sets and tuples. Such structures allow great flexibility in data representation.

Buneman, Davidson, Fernandez, Hillebrand and Suciu [7, 6, 15] propose a simple language UnQL for querying data organized as a rooted, edge-labeled graph. In this model, relational data may be represented as fixed-depth trees, and on such trees UnQL is equivalent to the relational algebra. The novelty of UnQL consists in its programming constructs for arbitrarily deep data and for cyclic structures. While strictly more powerful than query languages with path expressions like XSQL, UnQL can still be efficiently evaluated. We describe new optimization techniques for the deep or "vertical" dimension of UnQL queries. Furthermore, they show that known optimization techniques for operators on flat relations apply to the "horizontal" dimension of UnQL.

Fernandez, Popa and Suciu [14] have proposed a method of storing and querying semi-structured data, using storage schemas, which are closely related to recently introduced graph schemas. A storage schema splits the graph's edges into several relations, some of which may have labels of known types (such as strings or integers) while others may be still dynamically typed. They show that all positive queries in UnQL, a query language for semistructured data, can be translated into conjunctive queries against the relations in the storage schema. This result may be surprising, because UnQL is a powerful language, featuring regular path expressions, restructuring queries, joins, and unions.

Path constraints This class of constraints has been proposed for semistructured data to generalize integrity constraints that are found in traditional database management systems. Implication problems have been investigated by Buneman, Fan and Weinstein [16]. They characterized a schema in M in terms of a type constraint and an equality constraint, and investigate the interaction between these constraints and word constraints. They show that in the presence of equality and type constraint, the implication and finite implication problems for word constraints are also decidable, by giving a small model argument.

Looking at differences between semi-structured and structured data, one is tempted to think that adding structure simplifies reasoning about path constraints. Surprisingly, this is not the case. In the same paper it is shown that there is a fragment of the previosuly considered language whose associated implication and finite implication problems are decidable in PTIME, but are undecidable in the presence of type constraint.

Descriptive complexity and parallel query compilation Suciu and Tannen have proposed a new framework for parallel processing of collections. Its theoretical justification is a characterization (over ordered models) of the complexity class NC in terms of a divide-and-conquer form of recursion on finite sets [19, 17]. In order to support the efficient parallel compilation

of expressive query languages, they have defined and implemented a high-level language called CoPa for parallel processing of nested sets, bags, and sequences (a generalization of arrays and lists), featuring a powerful form of parallelizable recursion. CoPa has a formal declarative definition of parallel complexity as part of its operational specification and it was used to prove that the compilation process (architecture-independent in its majority) preserves the asymptotic complexity of the code [18, 20]. This implementation has allowed them to conduct speedup and scaleup experiments on a LogP simulator for the cost of data communication, control communication, and local computations involved in the parallel implementation of query languages for object-oriented or object-relational databases [20].

References

[1] A. Kosky and Susan Davidson and Peter Buneman. Semantics of Database Transformations. In L. Libkin and B. Thalheim, editor, *Semantics of Databases.* Springer LNCS 1358, Feb 1998.

[2] V. Breazu-Tannen, P. Buneman, and S. Naqvi. Structural recursion as a query language. In *Proceedings of 3rd International Workshop on Database Programming Languages, Naphlion, Greece*, pages 9–19. Morgan Kaufmann, August 1991. Also available as UPenn Technical Report MS-CIS-92-17.

[3] V. Breazu-Tannen and R. Subrahmanyam. Logical and computational aspects of programming with Sets/Bags/Lists. In *LNCS 510: Proceedings of 18th International Colloquium on Automata, Languages, and Programming, Madrid, Spain, July 1991*, pages 60–75. Springer Verlag, 1991.

[4] Val Breazu-Tannen, Peter Buneman, and Limsoon Wong. Naturally embedded query languages. In J. Biskup and R. Hull, editors, *LNCS 646: Proceedings of 4th International Conference on Database Theory, Berlin, Germany, October, 1992*, pages 140–154. Springer-Verlag, October 1992. Available as UPenn Technical Report MS-CIS-92-47.

[5] P. Buneman, J. Crabtree, S.B. Davidson, G.C. Overton, V. Tannen, and L. Wong. Biokleisli. In S. Letovsky, editor, *Bioinformatics.* Kluwer Academic Publishers. to appear.

[6] Peter Buneman, Susan Davidson, Gerd Hillebrand, and Dan Suciu. A query language and optimization techniques for unstructured data. In *Proceedings of ACM-SIGMOD International Conference on Management of Data*, pages 505–516, Montreal, Canada, June 1996.

[7] Peter Buneman, Susan Davidson, and Dan Suciu. Programming constructs for unstructured data. In *Proceedings of 5th International Workshop on Database Programming Languages*, Gubbio, Italy, September 1995.

[8] Peter Buneman, Leonid Libkin, Dan Suciu, Val Tannen, and Limsoon Wong. Comprehension syntax. *SIGMOD Record*, 23(1):87–96, March 1994.

[9] Peter Buneman, Shamim Naqvi, Val Tannen, and Limsoon Wong. Principles of proramming with collection types. *Theoretical Computer Science*, 149:3–48, 1995.

[10] S. Davidson, C. Hara, and L. Popa. Querying an object-oriented database using cpl. In *Proc. of the Brazilian Symposium on Databases*, 1997.

[11] S. Davidson, C. Overton, V. Tannen, and L. Wong. BioKleisli: A digital library for biomedical researchers. *Journal of Digital Libraries*, 1(1), 1996.

[12] Kazem Lellahi and Val Tannen. A calculus for collections and aggregates. In E. Moggi and G. Rosolini, editors, *LNCS 1290: Category Thory and Computer Science Proceedings of the 7th Int'l Conference, CTCS'97*, pages 261–280, Santa Margherita Ligure, September 1997. Springer-Verlag.

[13] Leonid Libkin, Rona Machlin, and Limsoon Wong. A query language for multidimensional arrays: Design, implementation, and optimization techniques. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 228–239, Montreal, Canada, June 1996.

[14] Mary Fernandez and Lucian Popa and Dan Suciu. A Structure-Based Approach to Querying Semi-Structured Data. In *International Workshop on Database Programming Languages (DBPL)*, pages 136–159, Estes Park, Colorado, August 1997.

[15] Peter Buneman and Susan Davidson and Mary Fernandez and Dan Suciu. Adding Structure to Unstructured Data. In *International Conference on Database Theory*, pages 336–351. Springer LNCS 1, Jan 1997.

[16] Peter Buneman and Wenfei Fan and Scott Weinstein. Path Constraints on Semistructured and Structured Data. In *PODS*, Jun 1998.

[17] Dan Suciu and Val Breazu-Tannen. A query language for NC. In *Proceedings of 13th ACM Symposium on Principles of Database Systems*, pages 167–178, Minneapolis, Minnesota, May 1994. See also UPenn Technical Report MS-CIS-94-05.

[18] Dan Suciu and Val Tannen. Efficient compilation of high-level data parallel algorithms. In *Proceedings of 6th ACM Symposium on Parallel Algorithms and Architectures*, pages 57–66, June 1994.

[19] Dan Suciu and Val Tannen. A query language for nc. *Journal of Computer and System Sciences*, 55(2), 1997.

[20] Dan Suciu and Val Tannen. Copa: a parallel programming language for collections. Technical report, University of Pennsylvania, 1998.

[21] Susan Davidson and A. Kosky. WOL: A Language for Database Transformations and Constraints. In *Proceedings of the International Conference of Data Engineering*, pages 55–65, Apr 1997.